

## К ВОПРОСУ ОПРЕДЕЛЕНИЯ ВРЕМЕННОЙ СЛОЖНОСТИ АЛГОРИТМОВ ТРАССИРОВКИ ЛУЧЕЙ

*В статье рассмотрена общая методика определения временной сложности алгоритмов трассировки лучей и обобщенная архитектура рейтрейсинга. Оценки временной сложности алгоритмов трассировки лучей показывают большие временные затраты на их реализацию, что делает невозможным решение задачи синтеза фотореалистичных изображений в реальном времени без применения высокопроизводительных параллельных вычислительных систем аппаратной поддержки.*

*Ключевые слова: визуализация, изображение, синтез, трассировка лучей, ускоряющие техники, архитектура, временная сложность*

S.A. ZORI

Donetsk national technical university, Pokrovsk

## TO THE QUESTION OF RAY TRACING TIME COMPLEXITY DETERMINATION

The article describes the method of determination of the overall time complexity of ray tracing algorithms and generalized architecture of ray tracing. Estimates of the time complexity of ray tracing algorithms show time-consuming to implement them, which makes it impossible to solve the problem of photo realistic image synthesis in real time without the use of high-performance parallel computing hardware support.

*Keywords: visualization, image synthesis, raytracing, accelerating technology, architecture, time complexity*

### Постановка задачи и анализ исследований и публикаций

Повышение скорости и качества систем 3D- синтеза и визуализации на основе методов трассировки лучей является актуальной и перспективной научной задачей.

За период с появления первого алгоритма трассировки лучей Уиттеда появилось огромное число алгоритмов класса трассировки лучей, позволявших достичь самого высокого качества изображений. Несмотря на значительное число существующих методов решения проблемы глобальной освещенности, эта область продолжает активно исследоваться [1].

Следует заметить, что, обеспечивая очень высокое качество синтезируемых изображений, трассировка лучей как метод визуализации имеет большую временную сложность, и, как правило, применяется на финальных стадиях подготовки изображений (в большинстве случаев одна картинка может трассироваться в течение нескольких часов и даже суток [1, 2, 3, 4]). Программные движки, выполняющие трассировку (рей-трейсеры), нередко являются составной частью сложных систем геометрического моделирования, таких как, например, AliasPowerAnimator, Maya и др. [5], причем часто они используют закрытую алгоритмическую реализацию технологии и обязательную современную аппаратную поддержку синтеза на многоядерных центральных процессорах и графических мультипроцессорах.

При рассмотрении различных техник трассировки лучей представляет интерес вопрос определения их временной сложности. В литературе не удалось найти полноценных материалов, посвященных определению временной сложности процесса.

### Основные временные затраты рейтрейсинга

Основные временные затраты и сложности, связанные с визуальными ошибками методов трассировки лучей, напрямую связаны с двумя основными моментами:

1. Поиск пересечения луча с объектами сцены.
2. Вычисление глобальной освещенности и определение эффектов освещения.

Поиск пересечений луча с геометрическими объектами выполняется наиболее часто и всегда является узким местом в трассировке лучей (до 75% общего времени). Для ускорения решения этой задачи используют ускоряющие структуры [6, 7, 8].

Выполнена классификация ускоряющих техник трассировки лучей, дан анализ особенностей и характеристики основных используемых ускоряющих техник (рис. 1).

Главным направлением из рассмотренных ускоряющих техник при этом является ускорение процедур поиска пересечений, так как оно не влечет за собой упрощений, которые могут привести к геометрическим неточностям или ухудшению качества синтеза.

В [6, 9] показано, что использование иерархических ускоряющих структур позволяет осуществлять поиск пересечений более эффективно, тем не менее возникают дополнительные затраты памяти на хранение данных структур и времени на их перестройку.

В качестве ускоряющих техник для повышения эффективности трассировки лучей сегодня в основном используют BVH структуры с AABB из-за простоты и скорости реализации или kd-tree структуры с SHA [6-9]. В [9, 10] автором подробно рассмотрен подход к оптимизации ускоряющей техники поиска пересечений луча с AABB и исследованы его характеристики.

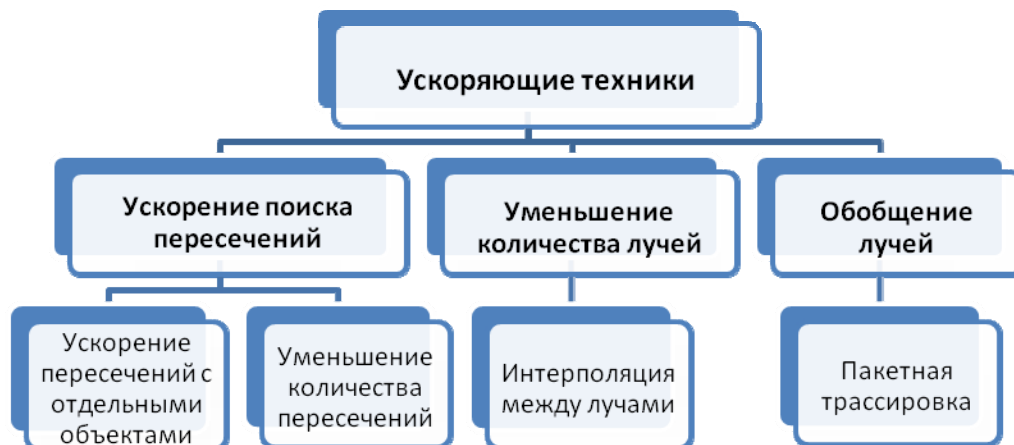


Рис. 1. Классификация ускоряющих техник трассировки лучей

Выполненный анализ теоретических основ определения глобальной освещенности для высококачественного фотореалистичного синтеза изображений методами трассировки лучей позволяет утверждать, что современные методы расчета глобальной освещенности основаны преимущественно на Монте-Карло интегрировании и алгоритме трассировки путей, который, несмотря на высокую вычислительную сложность, позволяет синтезировать большинство известных эффектов геометрической и волновой оптики [11]. Однако, их сверхвысокая временная сложность не позволяет проводить высококачественный синтез в реальном времени, и используются в основном для получения высококачественных изображений фотографического качества статических сцен [11, 12].

Вместе с тем, множество частных существующих реализаций этих методов показывают, что особенности освещения сцен и природа визуальных эффектов в специфических условиях могут быть использованы для внесения упрощений и значительного ускорения расчетов.

При рассмотрении различных техник трассировки лучей представляет интерес вопрос определения их теоретической временной сложности. В связи с этим предлагается обобщенная методика определения временной сложности алгоритмов трассировки лучей и общая архитектура рейтрейсинга в ней.

#### Временная сложность алгоритмов трассировки лучей

Рассмотренные методы и техники трассировки лучей позволяют выполнить обобщение процесса и определить временную сложность трассировки в общем случае.

Основные этапы синтеза изображений методами рейтрейсинга можно представить следующими этапами (рис. 2).

1. Создание модели и препроцессирование сцены (если необходимо).

2. Трассировка лучей.

2.1 Создание первичных лучей и определение пересечений лучей и объектов.

Для каждого пикселя экрана сформировать первичный луч и определить его пересечение со всеми объектами в сцене. Если пересечений больше, чем одно - выбрать ближайшее.

На этом этапе возможно применение ускоряющих техник.

2.2 Создание вторичных лучей и определение локального освещения

2.2.1 Если есть пересечение первичного луча с объектом - вычислить теневой луч (освещение) и лучи отражения и преломления.

Теневой луч проводится от точки пересечения к источнику(-ам) света. Его целью является определить, находится ли точка пересечения в тени конкретного источника света. Для каждого источника света должен быть свой теневой луч. Этот же теневой луч определяет локальное освещение (световой вектор), если луч на пути к источнику света не пересекается объектами сцены.

2.2.2 Рекурсивное повторение процедур 2.2.1 до достижения заданной степени точности или глубины пути.

2.3 Вычисление глобального освещения

После вычислений локального освещения в каждой точке пересечений лучей (п. 2.2), учитывать его на каждом следующем уровне дерева лучей (например, суммировать для прямого, отраженного и преломленного света с учетом соответствующих коэффициентов).

3. Запись цветов пикселей в буфер кадра или выходной файл.

Преобразовать полученные данные о цвете пикселя экранной плоскости в целочисленный формат представления цвета для дисплея.

С учетом рассмотренной архитектуры и основных этапов выполнения рейтрейсинга выполним оценку (в общем случае) временной сложности методов трассировки лучей (1):

$$T_{RT} = T_{scene_{pp}} + res_x \times res_y \times [sample_{pr} \times T_{pr} + sample_{sc} \times T_{sc}] + T_{im} \quad (1)$$

где  $T_{scene_{pp}}$  – время создания и подготовки (препроцессирования) модели сцены;  
 $res_x, res_y$  – разрешение экрана;  
 $sample_{pr}$  – количество генераций (сэмплов) первичных лучей;  
 $T_{pr}$  – время трассировки первичного луча;  
 $sample_{sc}$  – количество генераций (сэмплов) вторичных лучей;  
 $T_{sc}$  – время трассировки вторичного луча;  
 $T_{im}$  – время формирования изображения в буфере кадра.

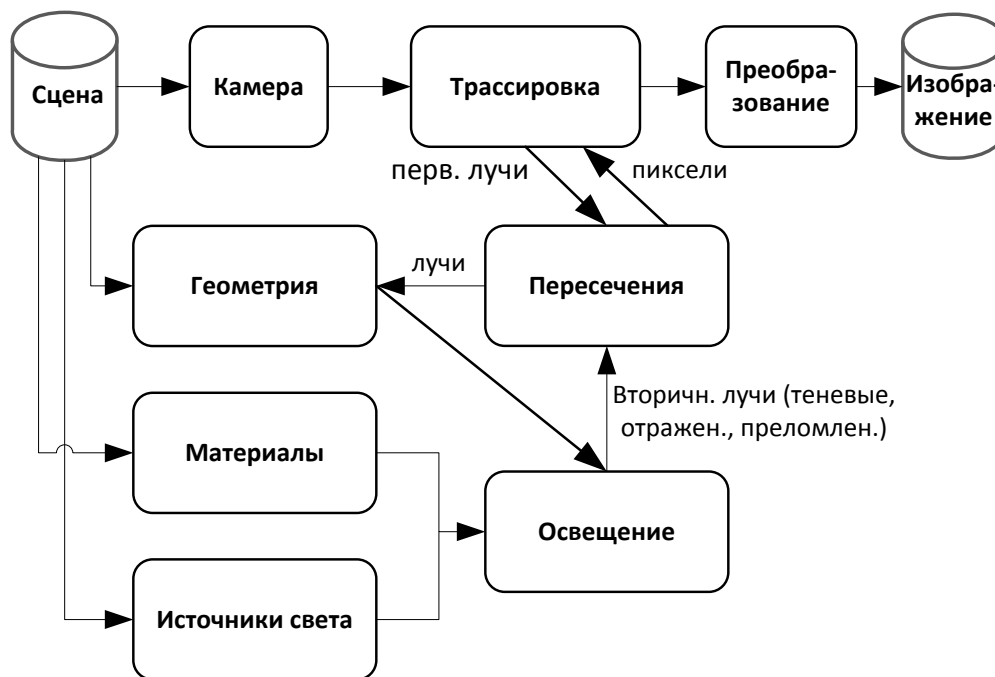


Рис. 2. Архитектура рейтрейсинга

Отметим, что количество сэмплов лучей определяется конкретным алгоритмом трассировки лучей (например, для алгоритма Уиттеда составляет 1, а для трассировки путей Монте-Карло для средне-статистической сцены и достижения приемлемого качества [13] – около 150 для первичных и 1500 для вторичных лучей).

Время формирования и препроцессирования сцены также зависит от конкретного алгоритма и применения ускоряющих структур. Например, препроцессирование при применении kd-tree с SHA:

$$T_{scene_{pp}} = N_o \times \log_2 N_o \times \tau_d \quad (2)$$

где  $N_o$  – количество объектов в сцене;

$\tau_d$  – время одной итерации подразделения пространства;

Время трассировки первичного луча можно определить как (3):

$$T_{pr} = T_{int_{pr}} + T_{lt\_sh_{loc}} \quad (3)$$

где

$T_{int_{pr}}$  – время нахождения пересечения первичного луча с объектами сцены;

$T_{lt\_sh_{loc}}$  – время вычисления локального освещения (световой вектор или теневой луч).

$T_{int_{pr}}$  и  $T_{lt\_sh_{loc}}$  определим следующим образом (4–5):

$T_{int_{pr}} = N_o \times \tau_{int}$  (без применения ускоряющих структур);

$$\log_2 N_o \times \tau_{int} \quad (\text{с применением kd-tree}) \quad (4)$$

где  $\tau_{int}$  – время вычисления пересечения луча с одним объектом.

$$T_{lt\_sh_{loc}} = N_l \times \tau_{lt\_sh} \quad (5)$$

где  $N_l$  – количество источников освещения в сцене;

$\tau_{lt\_sh}$  – время вычисления локального освещения в точке (световой или теневой луч).

Время  $T_{sc}$ , необходимое для трассировки вторичных лучей, можно определить следующим образом (6):

$$T_{sc} = T_{int_{sc}} + T_{lt_{sh_{loc}}} + l \times (T_{ref} + T_{sc}) \quad (6)$$

где  $T_{int_{sc}}$  – время нахождения пересечения вторичного луча с объектами сцены;

$T_{lt_{sh_{loc}}}$  – время вычисления локального освещения вторичного луча (световой вектор или теневой луч, определяется по (5));

$l$  – глубина трассировки пути;  $T_{ref}$  – время трассировки вторичных лучей преломления, определяется аналогично  $T_{sc}$ .

Как видно, в вычислениях вторичных лучей присутствует рекурсия.

Приведенные оценки временной сложности алгоритмов трассировки лучей явно показывают огромные временные затраты на их реализацию. И, даже с использованием модифицированных и оптимизированных методов трассировки, без применения высокопроизводительных параллельных вычислительных систем решение задачи синтеза в реальном времени не представляется возможным [1]. В последнее время появилось большое количество работ по адаптации алгоритмов рейтрейсинга под многопроцессорные системы [14] и графические процессоры [15, 16]. Известен ряд работ по разработке специализированных аппаратных трассировщиков лучей [17, 18].

Все это подтверждает актуальность дальнейших исследований и разработки способов ускорения алгоритмов рейтрейсинга как на алгоритмическом, так и на аппаратном уровне архитектур вычислительных систем.

### Выводы

В статье рассмотрена общая методика определения временной сложности алгоритмов трассировки лучей и обобщенная архитектура рейтрейсинга.

Приведенные оценки временной сложности алгоритмов трассировки лучей явно показывают большие временные затраты на их реализацию, что делает невозможным решение задачи синтеза фотореалистичных изображений в реальном времени без применения высокопроизводительных параллельных вычислительных систем аппаратной поддержки.

### Литература

1. Реалистическая пространственная визуализация с использованием технологий объемного отображения: Монография / Е.А.Башков, С.А.Зори. - Донецк, ГБУЗ "ДонНТУ", 2014. - 150 с.
2. Realistic Ray Tracing, Second Edition / P. Shirley, R. K. Morley. — Wellesley: A K Peters/CRC Press, 2003. — 235 с.
3. Jensen Henrik Wann, Christensen Per. High quality rendering using ray tracing and photon mapping // SIGGRAPH 2002 Course 43. New York, NY, USA: Association for Computing Machinery, Aug. 2002, ACM SIGGRAPH, 2002.
4. Akenine-Moller Real-Time Rendering. Third Edition / T. Akenine-Moller, E. Haines, N. Hoffman. — Wellesley: A. K. Peters, Ltd., 2008. — 1027 p.
5. Фролов В.А., Галактионов В.А., Трофимов М.А. Сравнительный анализ современных рендер-систем // Труды 24-й Международной конференции по компьютерной графике и зрению Графикон'2014. Южный Федеральный Университет, Ростов-на-Дону, Россия, 2014. С. 43–46.
6. Запорожченко И.А., Григорьев М.А., Зори С.А. Анализ методов уменьшения вычислительной сложности алгоритма трассировки лучей и способов их параллельной реализации / Моделювання та комп'ютерна графіка : Матеріали 5-ї міжнародної науково - технічної конференції, м. Донецьк, 24-27 вересня 2013 р. — Донецьк, ДонНТУ, Міністерство освіти та науки України, 2013. — С1.09, 3 с.
7. Foley Tim, Sugerman Jeremy. KD-tree acceleration structures for a GPU raytracer // Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. HWS '05. New York, NY, USA: ACM, 2005. p. 15–22.
8. Zhou, K., Hou, Q., Wang, R., and Guo, B. 2008. Real-time KD-tree construction on graphics hardware. In ACM SIGGRAPH Asia 2008 Papers (Singapore, December 10 - 13, 2008). J. C. Hart, Ed. SIGGRAPH Asia '08. ACM, New York, NY, p. 1-11
9. S.A. Zori, P.A. Porfiriov Productivity Increasing of Realistic Ray Tracing Stereo- Image Synthesis // Journal of Qafqaz University, Mathematics and Computer Science ISSN 1302-6763, Vol. 3, No 1, 2015.- p. 30- 38
10. Зори С.А. Объемная визуализация алгоритмом трассировки лучей с использованием двухуровневой иерархии ограничивающих объемов и AABB // Наукові праці Донецького національного технічного університету. Серія: "Інформатика, кібернетика та обчислювальна техніка" ISSN 1996-1588. №2 (21)' 2015.- с. 5-10
11. Фролов В.А. Методы решения проблемы глобальной освещенности на графических процессорах // Автореферат диссертации на соискание ученой степени: кандидат физико-математических наук.

Специальность: 05.13.11 - математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей. Москва, 2015 [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: <http://keldysh.ru/council/1/2015-frolov/avtoref.pdf>

12. Jensen, H. W. 2004. A practical guide to global illumination using ray tracing and photon mapping. In ACM SIGGRAPH 2004 Course Notes (Los Angeles, CA, August 08 - 12, 2004). SIGGRAPH '04. ACM, New York.

13. Фролов В., Фролов А. Трассировка путей (Path tracing) [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: <http://ray-tracing.ru/articles216.html>

14. F. Dachille and A. Kaufman. GI-Cube: An architecture for volumetric global illumination and rendering. In Proceedings of the 2000 SIGGRAPH/Eurographics Workshop on Graphics Hardware, p. 119–129, August 2000.

15. Garanzha K., Bely A., Premoze S., Galaktionov V. Out-of-core GPU ray tracing of complex scenes // In Technical talk Proceedings 38th International conference on computer graphics and interactive techniques ACM SIGGRAPH 2011, Article No.21, Vancouver, Canada – 2011.

16. Фролов Владимир, Игнатенко Алексей Интерактивная трассировка лучей и фотонные карты на GPU // В сборнике Proceedings of the 19th International Conference on Computer Graphics and Vision GraphiCon'2009, серия GraphiCon, MAKS Press Moscow, Russia, с. 255-262

17. Компания Siliconarts представила специализированный процессор для трассировки лучей [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: [http://www.oszone.net/18996/Siliconarts\\_develops\\_ray-tracing\\_graphics\\_processing\\_technology](http://www.oszone.net/18996/Siliconarts_develops_ray-tracing_graphics_processing_technology)

18. Imagination: трассировка лучей в реальном времени [Электронный ресурс]. – Электрон. текстовые данные. – Режим доступа: <http://www.3dnews.ru/797054>

#### References

1. 1. Realistichnaja prostranstvennaja vizualizacija s ispol'zovaniem tehnologiy ob'emnogo otobrajenija: Monografija / E.A.Bashkov, S.A.Zori. - Donetsk, GVUZ "DonNTU", 2014. - 150 s.

2. Realistic Ray Tracing, Second Edition / P. Shirley, R. K. Morley. — Wellesley: A K Peters/CRC Press, 2003. — 235 c.

3. Jensen Henrik Wann, Christensen Per. High quality rendering using ray tracing and photon mapping // SIGGRAPH 2002 Course Notes (Los Angeles, CA, August 08 - 12, 2004). SIGGRAPH '04. ACM, New York, NY, USA: Association for Computing Machinery, Aug. 2002, ACM SIG- GRAPH, 2002.

4. 4. Akenine-Moller Real-Time Rendering. Third Edition / T. Akenine-Moller, E. Haines, N. Hoffman. — Wellesley: A. K. Peters, Ltd., 2008. — 1027 p.

5. Frolov V.A., Galaktionov V.A., Trofimov M.A. Sravnitel'nyj analiz sovremennih render-sistem // GRAPHICON'2014. Ujyniy Federal'nyy Universitet, Rostov-na-Donu, Rossiya, 2014. s. 43–46.

6. Zaporozhenko I.A., Grigor'ev M.A., Zori S.A. Analiz metodov umen'shenija vychislitel'noy slojnosti algoritma trassirovki luchey i sposobov ih parallel'noy realizacii / Model'uvanja ta komp'uterna grafica: Materiali 5- mijnarodnoji naukovno- tehnicnoji konferenciji, Donetsk, 24-27.09.2013 — Donetsk, DonNTU, 2013. — S1.09, 3 s.

7. Foley Tim, Sugerman Jeremy. KD-tree acceleration structures for a GPU raytracer // Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. HWWS '05. New York, NY, USA: ACM, 2005. p. 15–22.

8. Zhou, K., Hou, Q., Wang, R., and Guo, B. 2008. Real-time KD-tree construction on graphics hardware. In ACM SIGGRAPH Asia 2008 Papers (Singapore, December 10 - 13, 2008). J. C. Hart, Ed. SIGGRAPH Asia '08. ACM, New York, NY, p. 1-11

9. S.A. Zori, P.A. Porfirov Productivity Increasing of Realistic Ray Tracing Stereo- Image Synthesis //Journal of Qafqaz University, Mathematics and Computer Science ISSN 1302-6763, Vol. 3, No 1, 2015.- p. 30- 38

10. Zori S.A. Ob'emnaja vizualizacija algoritmom trassirovki luchey s ispol'zovaniem dvuhurovnevoj ierarhii ogranichevajushih ob'emov i AABB // Naukovi pracj Donetskogo nacional'nogo tehnicnogo universitetu. Serija: "Informatika, kibernetika ta obchisl'uvajna tehnika" ISSN 1996-1588. №2 (21)' 2015.- s. 5-10

11. Frolov V.A. Metodi reshenija problemi global'noy osveshennosti na graficheskix processorax // Avtofegerat dissertacii na soiskanie uchenoy stepeni: kandidt fiziko-matematicheskix nauk. Special'nost': 05.13.11. Moskva, 2015 [Elektronnyy resurs]. – Rejim dostupa: <http://keldysh.ru/council/1/2015-frolov/avtoref.pdf>

12. Jensen, H. W. 2004. A practical guide to global illumination using ray tracing and photon mapping. In ACM SIGGRAPH 2004 Course Notes (Los Angeles, CA, August 08 - 12, 2004). SIGGRAPH '04. ACM, New York.

13. Frolov V., Frolov A. Trassirovka putey (Path tracing) [Elektronnyy resurs]. – Rejim dostupa: <http://ray-tracing.ru/articles216.html>

14. F. Dachille and A. Kaufman. GI-Cube: An architecture for volumetric global illumination and rendering. In Proceedings of the 2000 SIGGRAPH/Eurographics Workshop on Graphics Hardware, p. 119–129, August 2000.

15. Garanzha K., Bely A., Premoze S., Galaktionov V. Out-of-core GPU ray tracing of complex scenes // In Technical talk Proceedings 38th International conference on computer graphics and interactive techniques ACM SIGGRAPH 2011, Article No.21, Vancouver, Canada – 2011.

16. Frolov V., Ignatenko A. Interaktivnaja trassirovka luchey i fotonnye karti na GPU // Proceedings of the 19th International Conference on Computer Graphics and Vision GraphiCon'2009, serija GraphiCon, MAKS Press Moscow, Russia, s. 255-262

17. Kompanija Siliconarts predstavila specializirovannyj processor dlja trassirovki luchey [Elektronnyy resurs]. – Rejim dostupa: [http://www.oszone.net/18996/Siliconarts\\_develops\\_ray-tracing\\_graphics\\_processing\\_technology](http://www.oszone.net/18996/Siliconarts_develops_ray-tracing_graphics_processing_technology)

18. Imagination: trassirovka luchey v real'nom vremeni [Elektronnyy resurs]. Rejim dostupa: <http://www.3dnews.ru/797054>

Рецензія/Peer review : 25.5.2016 p.

Надрукована/Printed : 30.6.2016 p.

Стаття рецензована редакційною колегією